PKI Resource Query Protocol (PRQP)
draft-ietf-pkix-prqp-04

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 20, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Abstract

One of the most strategic problems still open in PKIX is locating public data and services associated with a Certification Authority (CA). This issue impacts interoperability and usability in PKIX.

This draft describes the PKI Resource Query Protocol (PRQP), its design, definition, and its impact in already deployed PKIX protocols.

Table of Contents

1. Requirements notation ........................................ 4
2. Introduction ................................................... 4
   2.1. Overview of existing solutions .......................... 4
       2.1.1. Certificate Extensions .................................. 4
       2.1.2. DNS SRV records ..................................... 5
       2.1.3. Local Network Oriented Solutions .................... 5
3. Protocol Details .................................................. 6
   3.1. The Resource Query Authority (RQA) ....................... 6
   3.2. PRQP Overview ............................................... 7
       3.2.1. PRQP Request ........................................ 7
           3.2.1.1. Request Syntax .................................. 8
       3.2.2. PRQP Response ....................................... 10
           3.2.2.1. Response Syntax ................................ 10
4. Object Identifiers for PKI resources ............................ 12
   4.1. The RQA Service identifier ............................... 13
   4.2. The OCSP identifier ...................................... 13
   4.3. The Subject’s Certificate identifier ..................... 13
   4.4. The Issuer’s Certificate identifier ...................... 14
   4.5. The Timestamping Service identifier ...................... 14
   4.6. The SCVP Server identifier ............................... 15
   4.7. The CRL Distribution Point identifier .................... 15
   4.8. The Certificates Repository identifier ................... 15
   4.9. The CRL Repository identifier ............................ 15
   4.10. The Cross Certificates Repository identifier ............. 16
   4.11. The CMC Gateway identifier .............................. 16
   4.12. The CMP Gateway identifier .............................. 16
   4.13. The SCEP Gateway identifier ............................. 17
   4.14. The HTML Gateway identifier ............................ 17
   4.15. The XKMS Gateway identifier ............................ 17
4.16. The Certificate Policy (CP) identifier ............... 17
4.17. The Certification Practice Statement (CPS) identifier . 18
4.18. The Endorsed Trust Anchors identifier ............... 18
4.19. The LOA Policy (LP) identifier ...................... 18
4.20. The Certificate LOA Modifier identifier ............... 19
4.21. The HTML Certificate Request Service identifier .... 19
4.22. The HTML Certificate Revoke Service identifier .... 19
4.23. The HTML Certificate Renew Service identifier .... 20
4.24. The HTML Certificate Suspend Service identifier .... 20
4.25. The HTML Certificate Recovery Service Identifier ... 20
4.26. The Grid Accreditation Body identifier ............... 20
4.27. The Grid Policy identifier ......................... 21
4.28. The Grid Distribution Update identifier ............... 21
4.29. The Grid Accredited CA Certificates identifier .... 21
4.30. The Apex Trust Anchor Update identifier ............... 22
4.31. The Trust Anchor Update identifier ................. 22
4.32. The CA Incident Report identifier .................. 22
4.33. The Private Resources identifier .................... 23
5. IANA Considerations .................................. 23
6. PRQP Design Rationale ................................ 23
6.1. Response Complexity ................................ 23
6.2. RQA’s URL distribution .............................. 24
6.3. Security Considerations ............................. 24
6.4. Time Validity ........................................ 24
6.5. Message Format ...................................... 25
7. Acknowledgments ........................................ 25
8. References .............................................. 25
8.1. Normative References ................................. 25
8.2. Non-Normative References ............................. 27
Appendix A. Transport Protocol Specifications for PRQP
   Messages .............................................. 27
   A.1. PRQP over HTTP .................................... 27
   A.1.1. Request .......................................... 27
   A.1.2. Response ......................................... 28
   A.1.3. Message Caching ................................ 28
   A.2. PRQP over Peer-to-Peer Network .................... 29
Appendix B. RQA address Retrieval .......................... 29
   B.1. DHCP Specifications ............................... 29
   B.1.1. PRQP Servers IPv4 Option for DHCPv4 ............ 30
   B.1.2. PRQP Servers IPv6 Option for DHCPv6 ............ 30
   B.1.3. DHCP Configuration ............................. 31
   B.1.4. DHCP Client Processing .......................... 32
   B.2. DNS SRV Records .................................. 32
   B.2.1. SRV Record Format for PRQP ..................... 32
   B.2.2. Example: PRQP enabled zone file ................. 33
Appendix C. PRQP ASN1.1 Specification ....................... 34
Author’s Address ........................................ 38
1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

An increasing number of services and protocols are being defined to address different needs of users and administrators of PKIs. With the deployment of new applications and services, the need to access information and services provided by Certificate Service Providers (CSPs) is critical. Currently Certification Authorities (CAs) barely publish access details on their official web sites, this includes URL of provided services and repositories.

Using the PRQP, resources provided by a CA can be automatically and securely discovered by an application.

2.1. Overview of existing solutions

Currently there are three options to find URLs providing access to PKI data:

- by including such data in certificate extensions
- by searching easily accessible repositories (e.g. DNS, local database, etc.)
- by adapting existing protocols (e.g. SLP)

2.1.1. Certificate Extensions

To provide pointers to published data it is possible to use the Authority Information Access (AIA) Subject Information Access (SIA) extensions defined by PKIX [RFC3280].

The former can provide information about services associated with the issuer of the certificate, while the latter carries information (inside a CA certificate) about offered CA services.

AIA and SIA extensions are static, i.e. not modifiable unless the certificate is re-issued. If a CA inserts the AIA extension into every certificate it issues, e.g., to identify the location of an OCSP responder, then changing that location would require re-issuance of all these certificates, a substantial barrier to such a change. If a CA certificate is self-signed and used as a trust anchor, then
re-issuing the certificate to change the content of the SIA extension, e.g., to reflect a change in the location of a time stamping server would be very disruptive. In closed PKIs, e.g., enterprises, use of these extensions may be replaced by manual configuration and management of this data via ad hoc means. Because of the centrally controlled nature of such environments, the static nature of SIA and AIA extensions is not a concern.

However in order to promote interoperability between PKIs, PRQP enables dynamic management of pointers to such services (e.g., adding/removing or moving) without requiring changes in the certificate contents or third parties to manually configure services in their applications. Even in closed environments, PRQP could help manage PKI services analogous the way DHCP facilitates network management.

2.1.2. DNS SRV records

The SRV record technique provides pointers to servers via the DNS [RFC1035].

As defined in [RFC2782], the introduction of this type of record allows administrators to perform operations similar to what we require in order to solve the problem we are addressing in this draft, i.e., to provide URLs to services.

The problem in the adoption of this mechanism is that, in contrast to the DNS environment, usually in PKIX there is no fixed mapping between certificates and the DNS name space. The only exception is when the Domain Component (DC) attributes are used in the certificate’s Subject.

Currently this approach is not widely adopted. Moreover, it is not always easy to identify the right DNS to query to, when trying to find a particular service provided by a CA, because of the lack of such information in certificates.

2.1.3. Local Network Oriented Solutions

Another approach to provide reliable information is to use existing protocols for service location such as Jini, Universal Plug and Play protocol (UPnP) or Service Location Protocol (SLP) [RFC2608] [RFC2609].

The IETF defined the SLP to provide a service location mechanism that is language and technology independent. Some issues, however, make it not the right choice to solve our problem, e.g., the protocol is quite complex to implement when considering the scope of the problem
we are addressing.

The definition of a specific and simple protocol for PKI service and resource location is needed to ease PKI integration into existing and future applications, especially for mobile devices which have limited computational power and communication bandwidth.

3. Protocol Details

The PRQP protocol is a request-response protocol, formed by the exchanging of two messages, i.e., a request and a response between a client and a server, called the Resource Query Authority (RQA).

The requesting entity (the client) may be any entity that needs to access information about repositories and services related to a certificate.

The RQA is the authority entitled to answer for a particular CA or to act as a PRQP Trusted Authority (PTA) for a set of users, e.g., users in an enterprise environment.

In the first case the RQA is directly designated by a CA to act as an RQA, by having the CA issue a certificate to the RQA with a specific value set in the extendedKeyUsage extension. In this case the RQA provides authoritative responses for requests regarding the CA that issued the RQA’s certificate.

When operating as a PTA, the RQA may provide responses about multiple CAs, without the need to have been directly certified by them. To operate as such, a specific extension (prqpTrustedAuthority) should be present in RQA’s certificate and its value should be set to TRUE.

3.1. The Resource Query Authority (RQA)

The Resource Query Authority is the designated authority to act as PRQP responder. The RQA’s signing key needs not to be the same as that of the CA that designated it.

The CA may designate an RQA by issuing a certificate containing a unique value for the extendedKeyUsage in RQA’s certificate. The RQA may also act as a trusted responder. PRQP signing delegation SHALL be designated by the inclusion of id-kp-PRQPSigning in the extendedKeyUsage extension within the PRQP response signer’s certificate.
id-kp-PRQPSigning OBJECT IDENTIFIER ::= {iso(1) identified-organization(2) dod(6) internet(1) security(5) mechanisms(5) pkix(7) kp(3) 11}

When operating as a PTA, the RQA may provide responses about multiple CAs, without the need to have been directly certified by them. To operate as a PTA a specific extension (prqpTrustedAuthority) should be present in RQA’s certificate and its value should be set to TRUE.

prqpTrustedAuthority ::= BOOLEAN DEFAULT TRUE

We also define two new OIDs to identify the PRQP protocol and the PTA extension as follows:

id-prqp OBJECT IDENTIFIER ::= { id-pkix 23 }

id-prqp-pta OBJECT IDENTIFIER ::= { id-prqp 1 }

3.2.  PRQP Overview

The protocol encompasses the exchange of a single round of messages between a client and an RQA:

1. the client requests a resource token by sending a request to the RQA

2. the RQA replies by sending a response to the client

Upon receiving the response the client MUST verify the status error returned in the response. If no error is present, the client MUST verify the various fields contained in the ResourceResponseToken and the validity of the associated digital signature (if present). A nonce MAY be used to guarantee that the response is associated with a specific request in order to avoid reply attacks.

The client also SHOULD check the validity period of the response. It SHOULD NOT, in order to minimize the load on an RQA, request again the location of the same resource within this interval to the same RQA.

If the response is signed, the client SHOULD check the RQA’s certificate validity.

3.2.1. PRQP Request

A PRQP request contains the following data:
3.2.1.1. Request Syntax

The PRQP request syntax is as follows:

```
PRQPRequest ::= SEQUENCE {
  requestData            TBSReqData,
  signature              [0] EXPLICIT Signature OPTIONAL }
```

```
TBSReqData ::= SEQUENCE {
  version                INTEGER { v(1) },
  nonce              [0] INTEGER              OPTIONAL,
  -- very large number
  producedAt             GeneralizedTime,
  -- time when the request has been generated
  serviceToken           ResourceRequestToken,
  -- token identifying the requested service
  extensions         [1] IMPLICIT Extensions  OPTIONAL }
```

The version field (currently v1) describes the version of the PRQP request. The nonce field, if present, is an integer between 80 bits and 256 bit in length. The producedAt define the time-frame when the request has been generated.

```
ResourceRequestToken ::= SEQUENCE {
  ca                      CertIdentifier,
  servicesList        [0] SET OF ResourceIdentifier OPTIONAL }
```

The ca field is of type CertIdentifier. This is used to identify the certificate of the CA whose services are requested.

The CertIdentifier syntax is as follows:
BasicCertIdentifier ::= SEQUENCE {
  issuerNameHash          OCTET STRING,
  serialNumber            CertificateSerialNumber  }

ExtendedCertInfo ::= SEQUENCE {
  certificateHash         OCTET STRING,
  subjectKeyHash          OCTET STRING,
  subjectKeyIdentifier    [0] KeyIdentifier OPTIONAL,
  issuerKeyIdentifier     [1] KeyIdentifier  OPTIONAL  }

CertIdentifier ::= SEQUENCE {
  hashAlgorithm           AlgorithmIdentifier,
  basicCertIdentifier     BasicCertIdentifier,
  extInfo                 [0] ExtendedCertInfo OPTIONAL,
  caCertificate           [1] Certificate OPTIONAL,

The resourceList specifies the resources or services being requested.

ResourceIdentifier ::= SEQUENCE {
  resourceId             OBJECT IDENTIFIER,
  version                 [0] INTEGER OPTIONAL
  --- version of the protocol or data format (if applicable)
  oid                     [1] OBJECT IDENTIFIER  OPTIONAL,
  --- object identifier associated with the URL
  --- (if applicable)  }

The ResourceIdentifier is formed by an OID that identifies the service or the data being requested (e.g. OCSP, LDAP, CRL, etc...) and an optional version number that may be used to better identify the requested resource. All fields SHOULD be used whenever applicable.

If one or more ResourceIdentifier are provided in the request, the RQA should report back the location for each of the requested services. If no ResourceIdentifier is present in the request, the response should carry all the available service locations for the specified CA (with respect to the MaxResponse and optional parameters constrain).

The signature field is of type Signature and it is defined in

[RFC2560]:

Signature ::= SEQUENCE {
  signatureAlgorithm     AlgorithmIdentifier,
  signature              BIT STRING,
  certs                  [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
Extensions can be used for future protocol enhancement.

### 3.2.2. PRQP Response

The PRQP response contains the following data:

- protocol version
- nonce
- status
- CA identifier
- ResourceResponseToken
- Extensions

#### 3.2.2.1. Response Syntax

The response syntax is as follows:

```plaintext
PRQPResponse ::= SEQUENCE {
  respData               TBSRespData,
  signature          [0] EXPLICIT Signature OPTIONAL }

TBSRespData ::= SEQUENCE {
  version                INTEGER { v(1) },
  nonce              [0] INTEGER              OPTIONAL,
    -- as duplicated from the request
  producedAt             GeneralizedTime,
    -- time when the response has been generated
  nextUpdate         [1] GeneralizedTime      OPTIONAL,
    -- time till when the response should be considered valid
  pkiStatus              PKIStatusInfo,
    -- status of the response
  caCertId               CertIdentifier,
    -- identifier of the CA certificate that issued the
    -- targeted certificate
    -- token carrying information about
    -- requested services
  extensions         [3] EXPLICIT Extensions  OPTIONAL }
```

The version field (currently v1) describes the version of the used PRQP response. The nonce, if present, binds the response to a specific request. The usage of the nonce is meaningful only in
signed responses and its value must be copied directly from the corresponding request. If not present in the request, the nonce MUST be omitted.

The pkiStatus field is used to return useful information to the client on the status of the query.

\[
\text{PKIStatusInfo ::= SEQUENCE } \{
\begin{array}{l}
\text{status PKIStatus, } \\
\text{statusString [0] UTF8String OPTIONAL, } \\
\text{failInfo [1] PKIFailureInfo OPTIONAL, } \\
\text{referrals [2] EXPLICIT SEQUENCE OF IA5String OPTIONAL}
\end{array}
\}
\]

If status has value zero, a responseToken MUST be present in the response. When the status value is non zero, the responseToken MUST be omitted and the reason code MUST be one of the values in PKIStatus. When the PKIStatus value is set to caNotPresent (2) or systemFailure (3), a list referral URLs MAY be included in the response to facilitate the client in finding the required resource from other known servers.

\[
\text{PKIStatus ::= INTEGER } \{ 
\begin{array}{l}
\text{ok (0), } \\
\text{badRequest (1), } \\
\text{caNotPresent (2), } \\
\text{systemFailure (3)}
\end{array}
\}
\]

The signature field is of type Signature and it is defined in [RFC2560]:

\[
\text{Signature ::= SEQUENCE } \{ 
\begin{array}{l}
\text{signatureAlgorithm AlgorithmIdentifier, } \\
\text{signature BIT STRING, } \\
\text{certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}
\end{array}
\}
\]

The responseToken carries information about the services requested by the client. For each of the requested service, the RQA should include a ResourceResponseToken which bears the OID of the service and the corresponding URI.

The ResourceResponseToken syntax is described below:
ResourceResponseToken ::= SEQUENCE {
  resourceId  OBJECT IDENTIFIER,  --- resource identifier
  resourceLocatorList [0] EXPLICIT SEQUENCE OF IA5String,  --- sequence of resource locators (URI)
  version       [1] INTEGER             OPTIONAL,  --- version of the protocol or data format (if applicable)
  oid           [2] OBJECT IDENTIFIER   OPTIONAL,  --- object identifier associated with the URL
                (if applicable)
  resourceInfo  [3] UTF8String           OPTIONAL,  --- additional service Info (eg. technical contacts) }

The resourceId field value is copied from the corresponding request and it bears the OID of the service about which the client inquired. In section Section 4 we define a list of default PKI resources.

The producedAt and nextUpdate define the time-frame when the response data is to be considered valid. Within the defined period, the client SHOULD NOT request for the same service. Use of wider time-frames values can help the RQA avoid duplication of requests from the same client thus potentially lowering the load of the responder. However, providing this data to a client does not ensure a lower query rate, as a server cannot rely on clients to obey the advice provided in the response.

The resourceLocator bears access information for the service identified by the serviceId. The name MUST be an absolute URL, and it MUST follow the URL syntax and encoding rules specified in [RFC4248] and [RFC4266]. The resourceLocator includes both a scheme (e.g., HTTP or FTP) and a scheme specific part. The scheme specific part is supposed to carry information on how to reach the requested service, this is, for example, a fully qualified domain name or IP address as the host. If the requested service is not available or it is unknown by the server, the resourceLocator value should be empty.

Optional Extensions may be added if requested.

4. Object Identifiers for PKI resources

The PRQP defines a set of standard OIDs that are used to identify resources related to a Certification Authority. In this section we provide a description for each of the defined OIDs.

The services are all defined under the id-ad-prqp OID which is defined as follows:
id-ad-prqp OBJECT IDENTIFIER ::= {id-ad 12}

4.1. The RQA Service identifier

When id-ad-prqp-rqa is used in a PRQP message, the associated value in the response is the location of the PRQP server (i.e., an RQA) using the conventions in this document or subsequent updates.

id-ad-prqp-rqa OBJECT IDENTIFIER ::= {id-ad-prqp 0}

The version field, if used, indicates the supported PRQP protocol version.

4.2. The OCSP identifier

When id-ad-prqp-ocsp appears in a PRQP request or response, the associated value in the response is the location of the OCSP responder, using the conventions defined in [RFC2560]. The version field, if used, indicates the supported protocol version.

id-ad-prqp-ocsp OBJECT IDENTIFIER ::= {id-ad-prqp 1}

4.3. The Subject’s Certificate identifier

When id-ad-subjectCert is used in a PRQP message, the associated value in the response is the location of the DER formatted certificate of the identified CA. The version field MAY be used to specify the version of the certificate pointed by the URL in a PRQP Response message. In order to enhance interoperability between applications and reduce development efforts, the URI should point directly to the certificate and not to a redirection service.

id-ad-prqp-subjectCert OBJECT IDENTIFIER ::= {id-ad-prqp 2}

HTTP server implementations accessed via the URI SHOULD specify the media type "application/x-x509-ca-cert" in the content-type header field of the response.

This field allows applications to check for renewal of CA certificates. When the application wants to check if a new version of the identified certificate exists, it can use this service and download the certificate from the URL. If the downloaded certificate differs from the one already possessed by the client, two different cases are possible:

1. The current certificate is not self-signed: in this case, the checks on the new certificate follow the rules specified in [RFC5280]. The new certificate can be safely added to the
application’s store (but not added to the list of Trusted Certificates or Trust Anchors) if it has been issued by the same issuer of the identified CA certificate.

2. The current certificate is self-signed: in this case, to avoid trust issues, the application should trust the pointed certificate only if the certificate has the same public key as the old one AND it is self signed (this provides proof of possession of the same private key).

For more complex trust anchor operations, please refer to Section 4.18, Section 4.30 or Section 4.31.

4.4. The Issuer’s Certificate identifier

When id-ad-issuerCert is used in a PRQP message, the associated value is in the response the location of the DER formatted certificate of the issuer of the identified CA. The version field MAY be used to specify the version of the certificate pointed by the URL in a PRQP Response message. In order to enhance interoperability between applications and reduce development efforts, the URI should point directly to the certificate and not to a redirection service.

   id-ad-prqp-issuerCert   OBJECT IDENTIFIER ::= {id-ad-prqp 3 }

HTTP server implementations accessed via the URI SHOULD specify the media type "application/x-x509-ca-cert" in the content-type header field of the response.

The content of this service is the same as the content of caIssuers when the provided URI refers to the CA Issuer’s certificate [RFC5280]. This field allows applications to dynamically download and build validation paths and may be extremely useful when cross certificates are used (eg., in bridge CAs).

4.5. The Timestamping Service identifier

When id-ad-timestamping is used in a PRQP message, the associated value in the response is the location of the Timestamping responder, using the conventions defined in [RFC3161]. The version field, if used, indicates the supported protocol version.

   id-ad-prqp-timestamping  OBJECT IDENTIFIER ::= {id-ad-prqp 4 }


4.6. The SCVP Server identifier

When id-ad-prqp-scvp appears in a PRQP request or response, the associated value in the response is the location of the SCVP responder, using the conventions defined in [RFC5055]. The version field, if used, indicates the supported protocol version.

   id-ad-prqp-scvp OBJECT IDENTIFIER ::= {id-ad-prqp 5 }

4.7. The CRL Distribution Point identifier

When id-ad-prqp-crlDistribution appears in a PRQP message, the associated value in the response is a pointer to the current CRL. The URI MUST point to a single DER encoded CRL as specified in [RFC2585]. The version field, if used, indicates the version of the pointed CRL. In order to enhance interoperability between applications and reduce development efforts, the URI should point directly to the CRL and not to a redirection service.

   id-ad-prqp-crlDistribution OBJECT IDENTIFIER ::= {id-ad-prqp 6 }

HTTP server implementations accessed via the URI SHOULD specify the media type "application/pkix-crl" in the content-type header field of the response.

4.8. The Certificates Repository identifier

When id-ad-prqp-certRepository appears in a PRQP message, the associated value in the response is a pointer to a set of certificates. The URI MUST point to a collection of certificates in a DER encoded "certs-only" CMS message as specified in [RFC5272].

   id-ad-prqp-certRepository OBJECT IDENTIFIER ::= {id-ad-prqp 7 }

HTTP server implementations accessed via the URI SHOULD specify the media type "application/pkcs7-mime" [RFC5272] in the content-type header field of the response. The name of the returned file SHOULD have a suffix of ".p7c" [RFC5272].

4.9. The CRL Repository identifier

When id-ad-prqp-crlRepository appears in a PRQP message, the associated value is a pointer to a set of CRL. The URI MUST point to a collection of CRLs in a DER encoded CMS message. The type of message should be a Simple PKI Response where the CRLs are placed in the CRL bag.

   id-ad-prqp-crlRepository OBJECT IDENTIFIER ::= {id-ad-prqp 8 }

Pala Expires May 20, 2010 [Page 15]
HTTP server implementations accessed via the URI SHOULD specify the media type "application/pkcs7-mime" [RFC5272] in the content-type header field of the response. The name of the returned file SHOULD have a suffix of ".p7c"

4.10. The Cross Certificates Repository identifier

When id-ad-prqp-crossCertRepository appears in a PRQP message, the associated value in the response is a pointer to a set of Cross Certificates. The URI MUST point to a collection of certificates in DER encoded CertificatePair object defined as:

CertificatePair ::= SEQUENCE {
   forward [0] Certificate OPTIONAL,
   reverse [1] Certificate OPTIONAL,
   -- at least one of the pair shall be present --
   }

The id-ad-prqp-crossCertRepository is defined as follows:

   id-ad-prqp-crossCertRepository
   OBJECT IDENTIFIER ::= {id-ad-prqp 9 }

As defined in [RFC4523], LDAP implementation store the CertificatePair in the crossCertificatePair attribute.

4.11. The CMC Gateway identifier

When id-ad-prqp-cmcGateway appears in a PRQP message, the associated value in the response is the location of the CMM over CMS service, using the conventions defined in [RFC5272]. As the [RFC5272] does not define a version for the protocol, if the version field is used to identify the service, applications SHOULD ignore it.

   id-ad-prqp-cmcGateway
   OBJECT IDENTIFIER ::= {id-ad-prqp 10 }

4.12. The CMP Gateway identifier

When id-ad-prqp-cmpGateway appears in a PRQP message, the associated value in the response is the location of the CMP over CMS service, using the conventions defined in [RFC4210]. As the [RFC4210] defines the protocol version in the pvno field of PKIHeader, the version field MAY be used to to identify the required/supported service version.

   id-ad-prqp-cmpGateway
   OBJECT IDENTIFIER ::= {id-ad-prqp 11 }
4.13. The SCEP Gateway identifier

When id-ad-prqp-scepGateway appears in a PRQP message, the associated value in the response is the location of the CMS service gateway, using the conventions defined in [I-D.nourse-scep]. The version field used to identify the service, if present, SHOULD be set to 0.

id-ad-prqp-scepGateway OBJECT IDENTIFIER ::= {id-ad-prqp 12 }

4.14. The HTML Gateway identifier

When id-ad-prqp-htmlGateway appears in a PRQP message, the associated value in the response is the location of a HTML CA service.

id-ad-prqp-htmlGateway OBJECT IDENTIFIER ::= {id-ad-prqp 13 }

The version field, if present, identifies the version of the HTML data as follows:

<table>
<thead>
<tr>
<th>Version Value</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HTML</td>
</tr>
<tr>
<td>1</td>
<td>XML</td>
</tr>
</tbody>
</table>

Table 1

4.15. The XKMS Gateway identifier

When id-ad-prqp-xkmsGateway appears in a PRQP message, the associated value in the response is the location of an XKMS server, using the conventions defined in [W3C.xkms] and [W3C.REC-xkms2-20050628]. The version field used to identify the service, if present, SHOULD be set to 1 for services compliant to [W3C.xkms] and to 2 for services compliant to [W3C.REC-xkms2-20050628].

id-ad-prqp-xkmsGateway OBJECT IDENTIFIER ::= {id-ad-prqp 14 }

4.16. The Certificate Policy (CP) identifier

When id-ad-prqp-certPolicy appears in a PRQP message, the associated value in the response is the location of a certificate Policy (CP). A CP may be used by a relying party to help in deciding whether a certificate, and the binding therein, are sufficiently trustworthy and otherwise appropriate for a particular application.

id-ad-prqp-certPolicy OBJECT IDENTIFIER ::= {id-ad-prqp 20 }
More information can be found in [RFC2527].

In order to gather the correct policy under which a certificate is issued, the optional OID field in the PRQPRequest SHOULD be copied from the certificate Policy extension of the EE certificate (i.e., the certificate issued by the CA the client is querying for).

4.17. The Certification Practice Statement (CPS) identifier

When id-ad-prqp-certPolicycertPracticeStatement appears in a PRQP message, the associated value in the response is the location of a Certification Practice Statement (CPS) published by the CA. A CPS is a document that details the practices and procedures established by a CA that will cover the life-cycle of certificates issued by the CA. That is it covers how the certificate will be generated, suspended and revoked. An internally focused document covering the internal environment of the CA.

id-ad-prqp-certPracticeStatement
OBJECT IDENTIFIER ::= {id-ad-prqp 21 }

More information can be found in [RFC2527].

4.18. The Endorsed Trust Anchors identifier

When id-ad-prqp-endorsedTA appears in a PRQP message, the associated value in the response is a pointer to a set of Trust Anchors (TA) in the form of certificates. The URI MUST point to a collection of certificates in a DER encoded CMS signedData message as specified in [RFC5272].

id-ad-prqp-endorsedTA OBJECT IDENTIFIER ::= {id-ad-prqp 22 }

HTTP server implementations accessed via the URI SHOULD specify the media type "application/pkcs7-mime" [RFC5272] in the content-type header field of the response. The name of the returned file SHOULD have a suffix of ".p7" [RFC5272]. The returned data object SHOULD be signed directly by the CA or by an authorized Identity whose certificate has been issued by the CA (i.e., an EE certificate). The application SHOULD verify the signature on the CMS message before proceeding in accepting the set of TAs. Moreover the application MAY import the set of certificates in its own certificate store as trusted depending on previous trust settings or input from the user.

4.19. The LOA Policy (LP) identifier

When id-ad-prqp-loaPolicy appears in a PRQP message, the associated value in the response is the location of a Level of Assurance Policy
(LP) published by the CA. An LP is a document that details the practices and procedures established by a CA that will cover the requirements for each Level of Assurance. The OID field in the request/response MAY be used to identify a specific LOA Policy document.

```plaintext
id-ad-prqp-loaPolicy OBJECT IDENTIFIER ::= {id-ad-prqp 25 }
```

More information can be found in [RFC2527].

### 4.20. The Certificate LOA Modifier identifier

When `id-ad-prqp-certLOAModifier` appears in a PRQP message, the associated value in the response is the location of a LOA Level Modifier. The LOA modifier service is used to identify the current LOA Level of the certificate (not the LOA under which the certificate has been issued).

```plaintext
id-ad-prqp-certLOAModifier OBJECT IDENTIFIER ::= {id-ad-prqp 26 }
```

### 4.21. The HTML Certificate Request Service identifier

When `id-ad-prqp-htmlRequestCertificate` appears in a PRQP message, the associated value in the response is the location of a HTML certificate request service. The version field, when present, identifies the version of the supported HTML format. See Table 1 for more details. As not standard exists that describes how to interact with a CA via HTML, this locator should be mainly used for browser-based certification requests.

```plaintext
id-ad-prqp-htmlRequestCertificate
OBJECT IDENTIFIER ::= {id-ad-prqp 30}
```

### 4.22. The HTML Certificate Revoke Service identifier

When `id-ad-prqp-htmlRevokeCertificate` appears in a PRQP message, the associated value in the response is the location of a HTML certificate revoking service. The version field, when present, identifies the version of the supported HTML format. See Table 1 for more details. As not standard exists that describes how to interact with a CA via HTML, this locator should be mainly used for browser-based certification requests.

```plaintext
id-ad-prqp-htmlRevokeCertificate
OBJECT IDENTIFIER ::= {id-ad-prqp 31}
```
4.23. The HTML Certificate Renew Service identifier

When id-ad-prqp-htmlRenewCertificate appears in a PRQP message, the associated value in the response is the location of a HTML certificate renewal service. The version field, when present, identifies the version of the supported HTML format. Table 1 As not standard exists that describes how to interact with a CA via HTML, this locator should be mainly used for browser-based certificate renewal requests.

\[
\text{id-ad-prqp-htmlRenewCertificate} \\
\text{OBJECT IDENTIFIER ::= \{id-ad-prqp 32\}}
\]

4.24. The HTML Certificate Suspend Service identifier

When id-ad-prqp-htmlSuspendCertificate appears in a PRQP message, the associated value in the response is the location of a HTML certificate suspension service. The version field, when present, identifies the version of the supported HTML format. See Table 1 for more details. As not standard exists that describes how to interact with a CA via HTML, this locator should be mainly used for browser-based certificate suspension requests.

\[
\text{id-ad-prqp-htmlSuspendCertificate} \\
\text{OBJECT IDENTIFIER ::= \{id-ad-prqp 33\}}
\]

4.25. The HTML Certificate Recovery Service Identifier

When id-ad-prqp-htmlRecoveryCertificate appears in a PRQP message, the associated value in the response is the location of a HTML certificate recovery service. The version field, when present, identifies the version of the supported HTML format. See Table 1 for more details. The recovery service is used when a user’s local copy of their keys and key history is destroyed. The recovery service returns the user to a complete state (e.g. so they can decrypt messages that were encrypted with older keys). As not standard exists that describes how to interact with a CA via HTML, this locator should be mainly used for browser-based certificate recovery requests.

\[
\text{id-ad-prqp-htmlRecoveryCertificate} \\
\text{OBJECT IDENTIFIER ::= \{id-ad-prqp 34\}}
\]

4.26. The Grid Accreditation Body identifier

When id-ad-prqp-grid-accreditationBody appears in a PRQP message, the associated value in the response is the location of the main information point of the Grid Policy Management Authority (GPMA) that
accredited the CA. The pointer SHOULD NOT be present if the CA has not been accredited by the GPMA.

id-ad-prqp-grid-accreditationBody
OBJECT IDENTIFIER ::= {id-ad-prqp 50}

4.27. The Grid Policy identifier

When id-ad-prqp-grid-accreditationPolicy appears in a PRQP message, the associated value in the response is the location of an Accreditation Policy published by a Grid Policy Management Authority (GPMA). The OID field SHOULD be used to uniquely identify the accreditation policy under which the CA has been accredited. The pointer SHOULD NOT be present if the CA has not been accredited by the GPMA. A Grid Policy (GP) is a document that details the practices and procedures required from a CA in order to be accredited by the GPMA.

id-ad-prqp-grid-accreditationPolicy
OBJECT IDENTIFIER ::= {id-ad-prqp 51}

4.28. The Grid Distribution Update identifier

When id-ad-prqp-grid-commonDistributionUpdate appears in a PRQP message, the associated value in the response is the location of the Grid Distribution Package associated with the Grid Policy Management Authority (GPMA) that accredited the CA. The OID field SHOULD be used to uniquely identify the accreditation policy under which the Grid Distribution Package has been released. The pointer SHOULD NOT be present if the CA has not been accredited by the GPMA.

id-ad-prqp-grid-commonDistributionUpdate
OBJECT IDENTIFIER ::= {id-ad-prqp 53}

4.29. The Grid Accredited CA Certificates identifier

When id-ad-prqp-gridAccreditedCACerts appears in a PRQP message, the associated value in the response is a pointer to a set of Trust Anchors (TA) in the form of certificates accredited by the Grid body/bodies that the CA is participating to. The URI MUST point to a collection of certificates in a DER encoded CMS signedData message as specified in [RFC5272]. The OID field SHOULD be used to uniquely identify the accreditation policy under which the set of CAs pointed by the URI have been accredited.

id-ad-prqp-grid-gridAccreditedCACerts
OBJECT IDENTIFIER ::= {id-ad-prqp 54}
HTTP server implementations accessed via the URI SHOULD specify the media type "application/pkcs7-mime" [RFC5272] in the content-type header field of the response. The name of the returned file SHOULD have a suffix of ".p7" [RFC5272]. The returned data object SHOULD be signed by the CA the endorsedTA service has been requested for. The application SHOULD verify the signature on the CMS message before proceeding in accepting the set of TAs. Moreover the application MAY import the set of certificates in its own certificate store as trusted depending on previous trust settings or input from the user.

4.30. The Apex Trust Anchor Update identifier

When id-ad-prqp-apexTampUpdate appears in a PRQP message, the associated value in the response is the location of a Apex Trust Anchor Update (apexTrustAnchorUpdate) message as defined by using the conventions defined in [I-D.pkix-tamp]. The version field used to identify the service, if present, SHOULD reflect the supported TAMP version.

    id-ad-prqp-apexTampUpdate             OBJECT IDENTIFIER ::= {id-ad-prqp 70}

4.31. The Trust Anchor Update identifier

When id-ad-prqp-tampUpdate appears in a PRQP message, the associated value in the response is the location of a Trust Anchor Update (trustAnchorUpdate) message as defined by using the conventions defined in [I-D.pkix-tamp]. The version field used to identify the service, if present, SHOULD reflect the supported TAMP version.

    id-ad-prqp-tampUpdate                OBJECT IDENTIFIER ::= {id-ad-prqp 71}

4.32. The CA Incident Report identifier

When id-ad-prqp-caIncidentReport appears in a PRQP message, the associated value in the response is the location of a Incident Report submission service. No standard mechanisms are currently defined for this type of service, therefore the The resourceInfo field in the response SHOULD be used to provide information on the provided Incident Report service. For example while the URI could point to a web-page carrying contacts information or a ticketing system for reporting CA-related incidents, the resourceInfo field could provide text carrying information that may be displayed to the user (e.g., a support phone number). This would allow support for a wide range of different devices and applications as long as they have the ability to display or read the content of the resourceInfo field to the user.

    id-ad-prqp-caIncidentReport          OBJECT IDENTIFIER ::= {id-ad-prqp 90}
4.33. The Private Resources identifier

When an application wants to identify private resources, i.e. services that are not standardized in the PRQP standard definition, id-ad-prqp-private should be used as the base OID:

```
id-ad-prqp-private OBJECT IDENTIFIER ::= {id-ad-prqp 100}
```

The OIDs for a private resource can be identified as follows:

```
myPrivateResource OBJECT IDENTIFIER ::= {id-ad-prqp-private N}
```

5. IANA Considerations

IANA has assigned a value of TBD1 for the DHCP option code described in Section Appendix B.1.1 of this document.

IANA has assigned a value of TBD2 for the DHCPv6 option code described in Section Appendix B.1.2 of this document.

6. PRQP Design Rationale

In this section we provide some considerations about the protocol design and its details.

6.1. Response Complexity

An important design consideration is the complexity of messages. Some type of services, e.g. delta CRLs, can be directly detected upon data downloading. On the contrary if a client is looking for a specific version of a protocol or data type, the definition of a fine-grained query system would allow for data downloading only when it is actually supported by the requesting client, thus reducing the server’s load.

At present we think that keeping the protocol simple will encourage its adoption in current environments because the flexibility introduced by PRQP is a big enhancement over the current options.

Moreover, without requiring changes to the protocol, extensions could be defined to provide more fine-grained options.

Future versions of the protocol may implement extended request and response types if required by applications.
6.2. RQA’s URL distribution

The AIA and SIA extensions in certificates can be used to carry the pointer to the RQA. If no RQA address is present in the certificate, a client application could use a default configured URL.

Although this approach seems to contradict the criticism of Certificate extensions use in Section 2.1.1, using only one extension to locate the RQA would provide an easy way to distribute the RQA’s URL.

The usage of PRQP will provide a gateway for all the other services and data URLs.

6.3. Security Considerations

The PRQP provides URLs for PKI resources. This means that it provides locators to data and services, not the data per se. It still remains the client’s job to access the provided URLs to gather the needed data.

Both NONCEs and signatures are optional in order to provide flexibility in how requests and responses are generated.

It is possible to provide pre-computed responses in case the NONCE is not provided by the client. This allows the RQA to generate off-line signatures for responses, an optimization used in OCSP.

Moreover if an authenticated secure channel is used at the transport level between the client and the RQA (e.g. HTTPS or SFTP) signatures in requests and responses can be safely omitted.

6.4. Time Validity

The time validity should reflect the frequency of updates in configured URLs. An interesting aspect to be considered is how often would users execute the protocol for a given set of data.

If the clients query the server often it could be a serious burden on the server but, if executed rarely, clients would not be able to discover changes in provided resources.

As described in more detail in Appendix A, the adoption of a validity time frame for responses can be used as a mean to balance the trade off between this two aspects, but this is merely advisory data for clients and thus not a guarantee against DoS attacks by clients.
6.5. Message Format

Two different candidates have been considered. The first one is the Extensible Markup Language (XML), while the second one is the Distinguished Encoding Rules (DER).

The adoption of the Abstract Syntax Notation (ASN.1) to describe the data structures allows a software developer to provide either DER or XML based implementations of the protocol.

However we think that a DER based implementation of PRQP is the best choice because of compatibility considerations with existing applications and APIs. Moreover DER encoded messages are smaller in size then XML encoded ones and almost all PKI aware applications already support it.

7. Acknowledgments

The authors would like to thank Stephen Kent for his insightful comments about PRQP and his help in writing this document.

8. References

8.1. Normative References

[I-D.ietf-dhc-option-guidelines]

[I-D.pkix-tamp]


8.2.  Non-Normative References

[I-D.nourse-scep]


[W3C.REC-xkms2-20050628]

[W3C.xkms]

Appendix A.  Transport Protocol Specifications for PRQP Messages

A.1.  PRQP over HTTP

This section describes the formatting needed in order to route PRQP request and response over HTTP.

A.1.1.  Request

HTTP based PRQP requests SHOULD use the POST method to submit their requests.  Where privacy is a requirement, PRQP transactions exchanged using HTTP MAY be protected using either TLS/SSL or some other lower layer protocol.
The required HTTP headers for the request are:

- Content-Type
- Content-Transfer-Encoding
- Content-Length

The Content-Type header SHOULD be set to "application/prqp-request". The Content-Transfer-Encoding SHOULD be set to "Binary", while the Content-Length SHOULD be set to the length (in bytes) of the body of the request. The body of the HTTP message MUST carry the binary value of the DER encoding of the PRQPRequest.

A.1.2. Response

An HTTP-based PRQP response is composed of the appropriate HTTP headers, followed by the binary value of the DER encoding of the PRQPResponse.

The required HTTP headers for the response are:

- Content-Type
- Content-Transfer-Encoding
- Content-Length

The Content-Type header SHOULD be set to "application/prqp-response". The Content-Transfer-Encoding SHOULD be set to "Binary", while the Content-Length SHOULD be set to the length (in bytes) of the body of the request. The body of the HTTP message MUST carry the binary value of the DER encoding of the PRQPResponse.

A.1.3. Message Caching

To minimize bandwidth usage, clients MUST locally cache authoritative PRQP responses for the validity period of the request. To enable proxy servers to be able to cache responses as well, additional HTTP headers MAY be used in the response.

The PRQP responder MAY ease caching by setting the following headers:

- date
- last-modified
In particular, the date field SHOULD carry the date at which the HTTP response has been generated. The last-modified, instead, SHOULD bear the date at which the response has been modified. This field SHOULD carry the same date as the producedAt field of the PRQPResponse. The expires field SHOULD carry the date till when the response is to be considered valid. This field SHOULD carry the same date as in the nextUpdate field of the PRQPResponse.

An example HTTP response would look like:

```
HTTP/1.0 200 OK
Content-Type: application/prqp-response
Content-Transfer-Encoding: Binary
Content-Length: 860
Date: Thu, 03 May 2007 04:43:43 GMT
Last-Modified: Thu, 03 May 2007 04:43:42 GMT
Expires: Thu, 04 May 2007 04:43:42 GMT

<...response data...>
```

PRQP clients SHOULD NOT included a no-cache header in PRQP request messages, unless the client encounters an expired response which may be a result of an intermediate proxy caching stale data.

### A.2. PRQP over Peer-to-Peer Network

PRQP offers a starting point for the development of a PKI Resource Discovery Architecture where different RQAs cooperate to access data not locally available.

One technology that already provides good results in data sharing is Peer-to-Peer (P2P) networking.

Signed PRQP requests and responses can be routed also on existing P2P networks or a PRQP-specific network can be setup to provide a World Wide PKI Resources Discovery Architecture (PRDA), the definition of which is out of the scope of this document. An example of such an architecture is PEACH [PEACH]

### Appendix B. RQA address Retrieval

#### B.1. DHCP Specifications

This section describes the needed steps to distribute RQAs addresses by using DHCP extensions. In particular we define the DHCP option
needed to identify an RQA server and we suggest options parsing for DHCP server and clients.

B.1.1. PRQP Servers IPv4 Option for DHCPv4

We define a prqp-servers option for DHCPv4 that specifies a list of Resource Query Authorities (PRQP servers) available to the client. The RQA address MUST be expressed as IPv4 addresses. Servers SHOULD be listed in order of preference and clients MUST treat the list of PRQP servers as an ordered list.

The format for the prqp-servers option is as shown below:

<table>
<thead>
<tr>
<th>Code</th>
<th>Len</th>
<th>Address 1</th>
<th>Address 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>n</td>
<td>a1</td>
<td>a2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a3</td>
<td>a4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a1</td>
<td>...</td>
</tr>
</tbody>
</table>

The code for the pki resource query authority list option is TBD1. The minimum length for this option is 1 octets.

B.1.2. PRQP Servers IPv6 Option for DHCPv6

We define a prqp-servers option for DHCPv6 that specifies a list of Resource Query Authorities (PRQP servers) available to the client. The RQA address MUST be expressed as IPv6 addresses (128-bit). Servers SHOULD be listed in order of preference and clients MUST treat the list of PRQP servers as an ordered list.

The format for the prqp-servers option is as follows:
option-code:  OPTION_PRQP_SERVERS (TBD2)

option-len:  Length of the ‘PRQP servers’ field in octets; it must be a multiple of 16

PRQP server:  IPv6 address of Resource Query Authority (PRQP) Server

The RQA address(es) specified in the ‘PRQP server’ MUST be encoded as IPv6 addresses. The code for the prqp-servers option for IPv6 is TBD2. The minimum length for this option is 1 octets.

B.1.3.  DHCP Configuration

As reported in [I-D.ietf-dhc-option-guidelines], one of the most deployed DHCP package is the ISC DHCP, mostly written by Ted Lemon in cooperation with Nominum, Inc. and now maintained by Internet Systems Consortium, Inc. ("ISC"). In order to provide developers and system administrators with deployment guidelines, we provide example configurations for both the server and the client.

Below is a sample configuration for the pki resource query authorities option that can be added both the dhclient.conf (on clients) and dhcpd.conf (on servers) for IPv4:

    option prqp-servers code TBD1 = array of ip-address;

If your environment supports IPv6, you should provide the option as a list of IPv6 addresses as follows:

    option prqp-servers code TBD2 = array of ipv6-address;

In addition to this, in order for the server to pass on the
configuration to the clients, the following example configuration options could be used in the server’s configuration file (typically /etc/dhcpd.conf):

```
... 
subnet XXX.XXX.XXX.XXX netmask YYY.YYY.YYY { 
 ... 
  option prqp-servers   rqa.openca.org, 
  rqa3.dartmouth.edu, 
  rqa5.mydomain.org; 
} 
... 
```

**B.1.4. DHCP Client Processing**

In order to provide applications deriving their configuration parameters from values provided by this DHCP option, the dhcp client needs to format the on-the-wire bits in a more digestible one. In particular for the "prqp-servers" option, a configuration file should be created as:

```
/etc/pki.conf
```

where the list of addresses can be stored. An example of such a file is reported below:

```
queryauthority rqa.openca.org
queryauthority rqa3.dartmouth.edu
queryauthority 127.0.0.1
```

where each line has the format:

```
queryauthority <ADDRESS>
```

**B.2. DNS SRV Records**

This section describes the needed steps to distribute RQAs addresses by using DNS SRV records. In particular we define the format to use for the SRV records. As an example we also provide a sample zone file.

**B.2.1. SRV Record Format for PRQP**

The format for DNS SRV records MUST be compliant with [RFC2782]. In particular, in order to support PRQP, a DNS server MUST use the following:

```
_Service._Proto.Name TTL Class SRV Priority Weight Port Target
```

Pala Expires May 20, 2010 [Page 32]
Where:

- Service is the symbolic name of the desired service. This field MUST be set to "rqa"
- Proto is the symbolic name of the protocol to be used. This field MUST be set to "_tcp"
- Name is the domain this RR refers to, i.e. the hostname of the RQA server
- TTL has the standard DNS meaning, refer to [RFC1035] for more information.
- Class has Standard DNS meaning [RFC1035]. This field MUST be set to "IN"
- Priority is the priority of this target host. The allowed range of values is 0-65535 (16 bit unsigned integer in network byte order).
- Weight is used for server selection mechanism. The weight field specifies a relative weight for entries with the same priority. The allowed range of values is 0-65535 (16 bit unsigned integer in network byte order). A more detailed description of the Weight usage can be found in [RFC2782].

B.2.2. Example: PRQP enabled zone file

In this section we provide a sample zone file for the domain .openca.org. In this example we configure service records for three different RQAs.
Appendix C. PRQP ASN1.1 Specification

PRQP DEFINITIONS EXPLICIT TAGS ::=  

BEGIN  

-- EXPORTS ALL --  

IMPORTS  

-- Directory Authentication Framework (X.509)  
Certificate, AlgorithmIdentifier  
FROM AuthenticationFramework { joint-iso-itu-t ds(5)  
module(1) authenticationFramework(7) 3 }  

-- PKIX Certificate Extensions  
AuthorityKeyIdentifier, SubjectKeyIdentifier, KeyIdentifier,  
FROM PKIX1Implicit88 {iso(1) identified-organization(3)  
dod(6) internet(1) security(5) mechanisms(5) pkix(7)  
id-mod(0) id-pkix1-implicit-88(2) }  

CertificateSerialNumber, Extensions, id-kp, id-ad-prqp  
FROM PKIX1Explicit88 {iso(1) identified-organization(3)  
dod(6) internet(1) security(5) mechanisms(5) pkix(7)  
id-mod(0) id-pkix1-explicit-88(1) };
PRQPRequest ::= SEQUENCE {
  requestData TBSReqData,
  signature [0] EXPLICIT Signature OPTIONAL }

TBSReqData ::= SEQUENCE {
  version INTEGER { v(1) },
  nonce [0] INTEGER OPTIONAL,
  -- very large number
  producedAt GeneralizedTime,
  -- time when the request has been generated
  serviceToken ResourceRequestToken,
  -- token identifying the requested service
  extensions [1] IMPLICIT Extensions OPTIONAL }

ResourceRequestToken ::= SEQUENCE {
  ca CertIdentifier,
  servicesList [0] SET OF ResourceIdentifier OPTIONAL }

BasicCertIdentifier ::= SEQUENCE {
  issuerNameHash OCTET STRING,
  serialNumber CertificateSerialNumber }

ExtenderCertInfo ::= SEQUENCE {
  certificateHash OCTET STRING,
  subjectKeyHash OCTET STRING,
  subjectKeyIdentifier [0] KeyIdentifier OPTIONAL,
  issuerKeyIdentifier [1] KeyIdentifier OPTIONAL }

CertIdentifier ::= SEQUENCE {
  hashAlgorithm AlgorithmIdentifier,
  basicCertIdentifier BasicCertIdentifier,
  extInfo [0] ExtendedCertInfo OPTIONAL,
  caCertificate [1] Certificate OPTIONAL,

ResourceIdentifier ::= SEQUENCE {
  resourceId OBJECT IDENTIFIER,
  version [0] INTEGER OPTIONAL
  --- version of the protocol or data format (if applicable)
  oid [1] OBJECT IDENTIFIER OPTIONAL
  --- object identifier associated with the URL
  --- (if applicable) }

PRQPResponse ::= SEQUENCE {
  respData TBSRespData,
  signature [0] EXPLICIT Signature OPTIONAL }
TBSRespData ::= SEQUENCE {
  version                INTEGER { v(1)},
  nonce                  INTEGER              OPTIONAL,
    -- as duplicated from the request
  producedAt             GeneralizedTime,
    -- time when the response has been generated
  nextUpdate         [0] GeneralizedTime      OPTIONAL,
    -- time till when the response should be considered
    -- valid
  pkiStatus              PKIStatusInfo,
    -- status of the response
  caCertId               CertIdentifier,
    -- identifier of the CA the targeted certificate is
    -- issued from
  responseToken          SEQUENCE OF ResourceResponseToken
    -- token carrying information about
    -- requested services
  extensions         [0] EXPLICIT Extensions  OPTIONAL }

PKIStatusInfo ::= SEQUENCE {
  status        PKIStatus,
  statusString  [0] UTF8String     OPTIONAL,
  failInfo      [1] PKIFailureInfo  OPTIONAL,
  referrals     [2] EXPLICIT SEQUENCE OF IA5String
    OPTIONAL }

PKIStatus ::= INTEGER {
  ok                     (0),
    -- when the PKIStatus contains the value zero one or
    -- more responseToken is present
  badRequest             (1),
    -- the request is badly formatted
  caNotPresent           (2),
    -- the requested CA is not present
  systemFailure          (3)
    -- a system failure has occourred }

Signature ::= SEQUENCE {
  signatureAlgorithm     AlgorithmIdentifier,
  signature              BIT STRING,
  certs              [0] EXPLICIT SEQUENCE OF Certificate
    OPTIONAL }

ResourceResponseToken ::= SEQUENCE {
resourceId  OBJECT IDENTIFIER,  
   --- resource identifier
resourceLocatorList [0] EXPLICIT SEQUENCE OF IA5String,  
   --- sequence of resource locators (URI)
version [1] INTEGER Optional,  
   --- version of the protocol or data format (if applicable)
oId [2] OBJECT IDENTIFIER Optional,  
   --- object identifier associated with the URL  
   (if applicable)
resourceInfo [3] UTF8String Optional,  
   --- additional service Info (eg. technical contacts)

-- Object Identifiers

id-kp-PRQPSigning  OBJECT IDENTIFIER ::= { id-kp 11 }
id-prqp  OBJECT IDENTIFIER ::= { id-pkix 23 }
id-ad-prqp  OBJECT IDENTIFIER ::= { id-prqp 1 }

id-ad-prqp  OBJECT IDENTIFIER ::= { id-ad 12 }
id-ad-prqp-rqa  OBJECT IDENTIFIER ::= { id-ad-prqp 0 }
id-ad-prqp-ocsp  OBJECT IDENTIFIER ::= { id-ad-prqp 1 }
id-ad-prqp-issuerCert  OBJECT IDENTIFIER ::= { id-ad-prqp 2 }
id-ad-prqp-timestamping  OBJECT IDENTIFIER ::= { id-ad-prqp 3 }
id-ad-prqp-scvp  OBJECT IDENTIFIER ::= { id-ad-prqp 4 }
id-ad-prqp-crlDistribution  OBJECT IDENTIFIER ::= { id-ad-prqp 5 }
id-ad-prqp-certRepository  OBJECT IDENTIFIER ::= { id-ad-prqp 6 }

id-ad-prqp-crossCertRepository  OBJECT IDENTIFIER ::= { id-ad-prqp 9 }
id-ad-prqp-cmcGateway  OBJECT IDENTIFIER ::= { id-ad-prqp 10 }
id-ad-prqp-cmpGateway  OBJECT IDENTIFIER ::= { id-ad-prqp 11 }
id-ad-prqp-scepGateway  OBJECT IDENTIFIER ::= { id-ad-prqp 12 }
id-ad-prqp-htmlGateway  OBJECT IDENTIFIER ::= { id-ad-prqp 13 }
id-ad-prqp-xkmsGateway  OBJECT IDENTIFIER ::= { id-ad-prqp 14 }

id-ad-prqp-certPolicy  OBJECT IDENTIFIER ::= { id-ad-prqp 15 }
id-ad-prqp-certPracticesStatement  OBJECT IDENTIFIER ::= { id-ad-prqp 20 }
id-ad-prqp-endorseTA  OBJECT IDENTIFIER ::= { id-ad-prqp 21 }
id-ad-prqp-loaPolicy  OBJECT IDENTIFIER ::= { id-ad-prqp 25 }
id-ad-prqp-certLOALevel  OBJECT IDENTIFIER ::= { id-ad-prqp 26 }
id-ad-prqp-htmlRequestCertificate  OBJECT IDENTIFIER ::= { id-ad-prqp 30 }
id-ad-prqp-htmlRevokeCertificate  OBJECT IDENTIFIER ::= { id-ad-prqp 31 }
id-ad-prqp-htmlRenewCertificate  OBJECT IDENTIFIER ::= { id-ad-prqp 32 }
id-ad-prqp-htmlSuspendCertificate  OBJECT IDENTIFIER ::= { id-ad-prqp 33 }
id-ad-prqp-gridRecoveryCertificate  OBJECT IDENTIFIER ::= { id-ad-prqp 34 }
id-ad-prqp-gridAccreditationBody  OBJECT IDENTIFIER ::= { id-ad-prqp 50 }
id-ad-prqp-gridAccreditationPolicy  OBJECT IDENTIFIER ::= { id-ad-prqp 51 }
id-ad-prqp-gridAccreditationStatus OBJECT IDENTIFIER ::= {id-ad-prqp 52}
id-ad-prqp-gridDistributionUpdate OBJECT IDENTIFIER ::= {id-ad-prqp 53}
id-ad-prqp-gridAccreditedCACerts OBJECT IDENTIFIER ::= {id-ad-prqp 54}
id-ad-prqp-apexTampUpdate OBJECT IDENTIFIER ::= {id-ad-prqp 70}
id-ad-prqp-tampUpdate OBJECT IDENTIFIER ::= {id-ad-prqp 71}
id-ad-prqp-caIncidentReport OBJECT IDENTIFIER ::= {id-ad-prqp 90}
id-ad-prqp-private OBJECT IDENTIFIER ::= {id-ad-prqp 100}

Author’s Address

Massimiliano Pala
Dartmouth College
6211 Sudikoff PKI/Trust Lab
Hanover, NH 03755
US

Email: pala@cs.dartmouth.edu
URI: http://www.openca.org